BROWN BOXES, INC.
Electronic Computer Attachments
26 Concord Road, Bedford, MA 01730
Phone: (617) 275-0090

THE 32K & 64K QUICK BROWN BOX INSTRUCTIONS

By S. Brown Pulliam **Rev. 11 April 1988**

Copyright 1988 by Brown Boxes, Inc. Bedford, MA.

1.0 WARRANTY:

The QUICK BROWN BOX is designed, manufactured and sold by:

BROWN BOXES, INC., 26 Concord Rd., Bedford, MA. 01730, USA.

ONE YEAR REPAIR or REPLACEMENT: If the *QUICK BROWN BOX* has any electrical or mechanical failure within ONE YEAR of shipment, it will be replaced or repaired (at the option of Brown Boxes, Inc.) at no cost except postage one way.
Written material and program source listings may be copied for individual, non-commercial use only.

1.1 CAUTION: The memory contents of the *QUICK BROWN BOX*, although protected, can be readily altered. It is recommended that users insure the security of programs in memory by backing up all such software to floppy disk or tape. Brown Boxes Inc. assumes no liability for the loss or malfunction of any program stored in the *QUICK BROWN BOX*.

2.0 INTRODUCTION

The *QUICK BROWN BOX* plugs into the Expansion Port of your C64/C128. It behaves as part of your computer's normal memory and you can store up to 32 or 64K of Basic and machine language programs in it, but when you turn off the computer, your programs don't disappear. They are stored unchanged, unless you wish to change them, for as long as 10 years. *QBB* memory contents are instantaneously available when you turn on your computer. Your cartridge can be plugged into any C64/128. The *QUICK BROWN BOX* is supplied already loaded with several utility programs that greatly simplify the storage of programs in the cartridge.

2.0.1 Commodore designed both the C64 and the C128 to allow the use of plug in cartridges. In general, the cartridges are used in place of some of the internal computer memory, and are limited to the addresses that the Commodore operating system allows them to use. An experienced programmer can easily structure his/her programs to reside at the appropriate memory address in the cartridge. Programs that have already been written to reside in other areas of memory that are not normally accessible by cartridge can be stored in the *QBB* and moved to their normal location when wanted. We have included Manager utilities in the *QBB* to take care of this kind of storage. When a program stored by the Manager is wanted, a couple of key strokes are all that are required to move the program out of the box, into its normal location, and have it running, all within a fraction of a second.

2.0.2 There is no theoretical limit to the number of programs that may be stored in the *QBB* at one time, but the total size of all programs combined may not exceed the capacity of the box, about 128 disk blocks for the 32K box, 255 for the 64K box. The *QBB* Manager allows a maximum of 30 program entries in the cartridge Directory. This puts a practical upper limit to the number of programs unless the user writes his/her own storage utility. Some commercial programs are incompatible with the *QBB* because they require some of the same memory address space that the box would use, without making provision to switch the box in or out as needed. It may still be possible to store such programs in

the *QBB*, and manually switch the box out before running the program.

2.0.3 The *QBB* is compatible with the Commodore 1700, 1750, and 1764 RAM Expansion Units provided a suitable socket expander is available to allow both devices to be connected to the computer expansion port at the same time. Refer to Section 9 for information about multiple cartridge operation.

2.1 INSTALLATION and OPERATION

As with any other cartridge:
PLUG IT IN ONLY WHEN THE COMPUTER IS OFF. LIKE MOST COMPUTER ATTACHMENTS THE QBB CAN BE HURT BY STATIC ELECTRICITY. WHEN NOT IN USE, KEEP IT IN THE ANTI-STATIC BAG IT CAME IN. BEFORE INSERTION, DISCHARGE YOUR ACCUMULATED STATIC BY TOUCHING THE REAR TV CONNECTOR.

SWITCHES: On the face of the cartridge are a 2 position slide switch and a push button switch. The slide switch selects either the C64 or the C128 mode of operation. If you are using the cartridge with the Commodore 64 computer, always keep the switch in the C64 position. Placing the slide switch in the 128 position will make the cartridge invisible to the computer. When using the C128 computer, the switch should be placed in whichever mode you intend to use the computer. If the switch is in the C64 position, it forces the C128 to operate in the C64 mode. The *QBB* that you receive is set up as a cartridge for the C64. **LEAVE THE SWITCH IN THE 64 POSITION UNTIL YOU HAVE FINISHED REVIEWING THESE INSTRUCTIONS AND HAVE SAVED THE UTILITIES TO DISK.**

2.1.1 The push button RESETS the computer and initializes the *QBB* to a Read Only mode. This RESET is useful if a crash leaves you with a non functioning keyboard. It allows you to regain control without losing program material you may have entered into the computer. If the cartridge is used with the C128, the push button function is identical to the C128 RESET.

2.1.2 The *QBB* has a Check Sum utility to allow you to verify that once you have loaded a particular combination of programs or data into the box, the box contents have not changed. When we ship the box, we do a check sum and write the number on the sticker tag you will see on the back of the box. To be sure the box contents have not been impaired during shipment, run the Check Sum program as soon as you receive your *QBB* . Make sure your computer is turned off, then plug the box in. Set the slide switch on the *QBB* to the 64 position (even if you are using it in a C128), turn on the computer, and a menu should be displayed on the screen. Enter *H then press RETURN and the check sum number should be displayed on the screen. If the number is correct, you should immediately save the entire box contents to a disk as described in Section 2.2.

2.1.3 If the cartridge appears to have a problem when it is turned on, (no menu appears or the wrong checksum is displayed), there are a few simple tests you can make that may save having to send the cartridge back. Refer to Section 11.

2.2 SAVING INSTALLED UTILITIES

Skilled programmers can figure out many ways to store programs and data in the *QUICK BROWN BOX*. Most of us, however, would prefer to have some

help. We have included in the *QBB* a number of programs to do this and they MUST be saved to disk to prevent their loss as you begin to experiment with ways of using your cartridge.

2.2.1 At the very least, you must save the entire box contents as a single file, and save two Managers and the *QBB* Loader as separate files. One of the Managers is for C64 boxes, and the other is for C128. They are used to initialize the *QBB* and begin building a new collection of programs in the box. They are also necessary whenever you wish to save the cartridge contents to disk, either as a packaged collection in a single sequential file, or as separate programs. The Manager **CANNOT** be used to reload the *QBB* with a previously packaged collection of programs, nor can it be used on any box unless it initialized that box. This precludes having the *QBB* store both C64 and C128 programs at the same time.

2.2.2 The *QBB* Loader program **MUST** be used to load a previously saved 'box' back into the cartridge. Unlike the two Managers, the same loader is used to load either a C64 or a C128 box. It must be used in the C64 mode. If it is used to load a collection of C128 programs, after the disk drive has completed the load, the Slide Switch is placed in the 128 position, and then a push on the RESET button is all that is required to activate it as a C128 box.

2.2.3 A quick introduction to the operation of the box and the terminology used in this manual will be helpful before you attempt the save. 'Enter' means type in the following instructions. Unless otherwise noted, an entry of an instruction is followed by use of the RETURN key.

1. Make sure your disk is formatted and has at least 200 free blocks. Put it into the drive.
2. Put the slide switch of the *QBB* in the 64 position.
3. With the computer turned OFF, plug the *QBB* in, then turn computer ON. The *QBB* Directory should then become visible.
4. Enter *M and the MANAGER.64 menu should be displayed.
5. Key the letter S and signify your wish to save the whole box by then keying the letter A.
6. You will be asked for a file name to be used on the disk directory. Enter a name and hit RETURN. It will take about a minute for the contents of the box to be saved to disk.
7. When the whole box save is done, you will be asked to hit RETURN to get back to the Manager menu. Then key the letter S again, this time signifying your wish to save a single program by pushing the letter B.
8. When you are asked for the program's *QBB* identifier, key in LD and when asked for the name that will be used in the disk directory, enter *QBB LOADER* and the loader will be saved to disk. At this point you have saved enough to restore the original box contents if anything goes wrong.
9. Next, again push S for SAVE, and, again as a single program, save MR using the disk file name C64MANAGER. This gives you the necessary starting point for building a new C64 box.
10. Finally, press the S for SAVE single program, and save M8 as disk file name C128MANAGER. This is your starter for any C128 box.

You now have saved the important utilities. The remaining utilities are saved as individual programs as was done in steps 7-8 above, AB is

the identifier for AUTOBOOT, DC the ident for MANAGER.DOC, HH is the CHECKSUM, SM is SUPERMON, and BC is the *QBB* Checker.

2.2.4 When all the programs have been saved to disk you can use the Manager program to delete those that are not immediately useful and to load new C64 programs into the *QBB* from disk, or you can start from scratch and set up a completely new cartridge. If you are going to set up a C128 *QBB*, you will have to start by loading the C128MANAGER from disk, with the computer in the 128 mode, slide switch in the 128 position, and then initialize the box.

3.0 USING the UTILITIES to LOAD PROGRAMS into the QBB

Now you are ready to store your own programs in the box. If your main application for the *QBB* is to save and run programs that are already written, most of the information you need will be in this section. If you intend to write your own programs for use in the box, the additional sections are provided to give information about the inner workings of the *QBB*.

3.1 USING THE MANAGERS

The Managers make saving and running programs ridiculously easy. Any Basic or machine language program that is not copy protected may be loaded from Disk or computer RAM into the *QBB*. Once in the box, a program may be moved into its normal running address by the entry of a few key strokes, and can start as soon as loaded, or be left dormant until a RUN or SYS command is entered. In addition, one program stored in the *QBB* can be designated the AUTOSTART program, and once that is done, it will load and RUN whenever the computer is turned on, with no user intervention whatsoever. The Manager will perform various ancillary functions, such as displaying the size of each program and the remaining box space, reading out a disk directory, viewing the box directory, or deleting individual programs or the entire box contents. It will also allow you to save individual programs from the box back to disk, or to save the entire contents of the box to disk as one complete sequential file. The one important thing the Manager won't do is put a previously packaged collection of programs back into the cartridge. That function is handled by the *QBB* Loader which is described in section 3.5.

3.1.1 The C64 and C128 Managers are very similar, so all the functions they have in common will be described first. The managers are not interchangable. You cannot have both C128 and C64 programs in the same cartridge at the same time. You must use the appropriate Manager to initialize the *QBB* and continue to use only that manager for all program additions or deletions until you choose to reinitialize the cartridge with the other Manager in the other mode. Before attempting to load any new programs in the *QBB*, make sure the important utilities have been saved to disk as described in Section 2.2.

3.1.2 Decide whether you wish to build a C64 or a C128 box. The type of computer, or its mode, must agree with the type of box. Load the corresponding Manager from disk, with the slide switch in the appropriate position, and when the Manager is ready, enter RUN and the Manager menu will appear:

VIEW *QBB* DIRECTORY (of stored programs)
LOAD NEW PROGRAM (from disk into *QBB*)
DELETE PROGRAM (from *QBB*)
INITIALIZE *QBB* (ready for all new)
SAVE *QBB* TO DISK (entire box or single programs)
AUTO-START PROGRAM (selects to run at power up)
READ DISK DIRECTORY
CHANGE PROGRAM DESCRIPTION
PARTITION *QBB*
EXIT (back to Basic READY)

The first letter of each menu line, when keyed on the keyboard, will initiate that action.

3.1.3 Initialize the *QBB* whenever you want to delete everything and start the cartridge with an empty directory. When the *QBB* is initialized (press I), you are asked to indicate the size of the box. Key an A for 64K, B for 32K or C for 16K. You can specify a smaller size than the actual box capacity. If you do, the Manager will only put programs in the portion you have specified, leaving the upper banks reserved for your own use. You will also be asked whether you wish to save the current contents to disk before erasing it. A final query is whether you wish the "RAM DISK" feature. This feature is useful if you are typing in or modifying a program or wish to modify the box contents without having to use the Manager. See more about that in para. 3.3

3.1.4 The box Directory is visible when you turn on the computer or push the RESET button. The READY prompt signifies that normal Commodore Basic is also available. You will note that in the Directory each program is listed by a two letter ident and up to six character name.

3.1.5 When you press V (View) the *QBB* directory appears on the screen. It also shows the number of disk blocks used by the programs in the box, and lets you know how many blocks are still available for your use. To delete a program from the box, you press D. You will then be asked to enter the two letter identifier for that program, and in a flash, the screen will appear, listing all of the remaining programs. If you call the M option again, you will see that the block count has been increased by the size of the deleted program. Use the R (Read) command to look at a disk directory to see what M.L. or Basic program you would like to load into the box. Then use the L (Load) command, which will prompt you to enter the name of the program as it was listed in the disk directory (without quotes). Use * as a wild card if you wish. If the program is too large to fit within the remaining space in the box, an error message will be displayed. Otherwise, it will be loaded into the box and you will be asked to supply a two letter identifier, then a name with a maximum of six characters for the *QBB* directory. The first character of the name can be used to mark the saved program for special handling. This is described in paragraph 3.1.8.

3.1.6 To have a program Auto-Start upon power up, use the A command, and you will be asked for the program's two letter identifier. Once that ID is entered, whenever the RESET button is pressed, or the computer turned on, that program will load and RUN. If you wish to run one of the other prorams in the *QBB*, usually just pushing the RESTORE key will call up the box directory. Some Auto-Start programs alter the

RESTORE vector, so if pushing the RESTORE key won't get you to the *QBB* directory, bypass the Auto-Start program by holding down the CONTROL key while powering up. When you see the normal Commodore startup screen, hit the RESTORE key a sharp stroke. If you wish to remove the Auto-Start designation from a program, you also use the A command.

3.1.7 The SAVE option may be used to save either a single program from the *QBB* back to disk, or to save the entire box contents as a single file. When a single entry is saved, it is saved in its normal program form. When the entire box is saved, it is saved as a Sequential file, and can only be subsequently loaded using the *QBB* Loader. Empty banks and partitioned banks will not be included in the save. Thus, the size of the sequential file depends only on those programs actually listed in the box directory.

3.1.8 You may not wish programs to run when called by the *ID. When you call a program, you may wish it to wait until you enter RUN or a SYS command. The program may be given this feature if when loading it using the Manager LOAD option, the first character of the six character name is the < symbol. This is particularly important if a C64 program you are loading into the *QBB* uses any of the memory between $8000 and $9FFF. Such a program would be in contention with the *QBB* for that area of memory. You would therefore wish to disconnect the *QBB* by moving the Slide Switch to the 128 position before running the program. Other programs may be linked and loaded when called from a program. If that is the intention, the first character should be a + sign. Further information about linked progams is contained in paragraph 3.6.

3.1.9 The CHANGE PROGRAM DESCRIPTION option will allow you to view the important identifier and memory address information concerning any program in the *QBB*. When it asks, you enter the two character identifier, and it then displays the six character name, the type (either Basic or M.L.), the load address, and the start address. If you wish to change name, type, or address you can do so by answering a series of questions. The C128 version has some differences, as described in Par. 3.2.2.

3.1.10 The PARTITION option allows you to set aside one or two 16K banks in the *QBB* for special uses. These banks will be protected from the Manager and from loading by the RAM Disk. If you attempt to partition a bank that already contains some programs, you will be given a warning and offered the option of deleting some programs to clear the bank for partitioning. The Manager won't partition into existing programs. The partitioned section of the box will always be the last bank or banks of the box.

3.2 USING THE C-128 MANAGER

The Manager options described for the C64 in section 3.1 also apply to the C128 Manager. The computer (and *QBB*) should always be in the 128 mode when using this Manager. Do not attempt to use the 128 Manager from the cartridge as you originally received it. Call it from the disk to which you have saved it, by entering DLOAD "MGR128". When it is loaded, enter RUN (RETURN), and it will then display its menu, which is similar to that of the C64 Manager. If this is your first attempt to use the cartridge in the 128 mode, the *QUICK BROWN BOX* must be 'initialized' by the C128 Manager, so you would begin by using the

INITIALIZE option.

3.2.1 Because of the multiple 'banks' of the C128, the LOAD option of the Manager will require additional information if you are loading a machine language program into the *QBB*. In addition to a two letter ID and a six letter name you will be asked whether you want to use the default Bank values for your program, Bank 0 (all RAM), where most programs load, and Bank 15 for the bank from which the program will be run (the SYS address). When in doubt, use the default banks. You can often determine where a progam should be located from the loading instructions that are found in magazine articles that accompany programs, or from studying the basic programs that are used to load machine language programs. the BLOAD command is usually followed by B0 or B1, the bank for the load, and SYS commands may be preceded by a Bank command. If neither is present, you can allow the Manager to keep the default values.

3.2.2 If a machine language progam doesn't run properly when it is called from the *QBB* by pressing * and its ID, you may have chosen the wrong Banks for the LOAD or SYS. You can use the CHANGE PROGRAM DESCRIPTION option of the Manager menu to select other banks. In additon to the ID, six character name, and program type, the 128 verson displays RAM banks (separated by a comma) and Load address. You can change the bank values to any decimal number from 0 to 15, but they must be separated by the comma. Use it as often as necessary. A more serious problem may arise if the program has been written to disk using the wrong load address information. It will not run proper- ly from the *QBB*, and no juggling of banks will correct the problem. The only solution is to determine what load address was intended if you can get access to the documentation, and revise the load address so it will run from the *QBB*.

3.2.3 As with the C64 Manager, the SAVE command saves a complete box to disk or saves individual programs. When saving a complete box, keep in mind that when you wish to re-load that file into the *QBB*, you will be using *QBB* Loader in the C64 mode. If the disk you are saving to is already fairly full, the file will be put on the second side, making it difficult to retrieve in the 64 mode. Therefore, try to do the save to the front side of the disk. Section 3.5 contains the instructions for using the Loader to restore a set of programs to the *QBB*.

3.3 RAM DISK

Normally, when you wish to store a program in the *QBB*, the Manager is used to move the program from disk into *QBB* RAM. If you are a programmer, and wish to save a progam you are working on to the *QBB*, you would have to save it to disk first, and then use the Manager to move it from disk to the *QBB*. The *QBB* RAM DISK commands free you from having to use the disk drive as an intermediary. With the RAM DISK activated, you can save the program directly to the *QBB*. This option should not be confused with the Commodore RAM DOS for their RAM Expansion Units. The *QBB* RAM DISK does not use regular disk commands. When RAM DISK is installed, you have the further Save/Replace option, which allows you to save a Basic program in a protected area in the *QBB* with two keystrokes. It is the next best thing to actually programming in non-volatile memory itself. The penalty you pay for using the RAM DISK is the loss of about 4 blocks of box storage space. If you also select

Save/Replace a further 65 blocks of *QBB* space is removed from any other use.

3.3.1 If you wish to be able to use the *QBB* RAM DISK (or Save/Replace) it must be activated when the cartridge is first initialized, before any programs have been stored. You may re-initialize the *QBB* as described in paragraph 3.1.7 and when doing so, answer with a Y when asked about RAM DISK, and a Y or N for SAVE/REPLACE.

3.3.2 With the RAM DISK activated, any time a program is in the computer memory, you may save it to the *QBB*. If the program is in Basic, you would enter the up arrow (↑) followed by a two letter identifier, then a name with a maximum of six characters within quote marks. If the program is in machine language the second quote must be followed by a comma then start address, comma, end address +1. Thus the form would be:

Basic: ↑ID "NAME"
M.L.: ↑ID "NAME",12345,54321

The numbers may be expressed in hexadecimal provided that the address values use four digits and are preceded by a $, and the bank values use two digits. For example Bank 13 could also be expressed as $0D. Mixing Decimal and Hex is O.K.
NOTE: In C64 mode, you cannot use the RAM DISK to save programs to the *QBB* from addresses $8000-$BFFF which is the RAM under the *QBB*.

3.3.3 If you are storing a machine language program in the C128 you may need to provide further information. In addition to the start and end address + 1, you may also indicate which C128 Bank you want the program to load to, and from which Bank it would be called to run (the SYS address). If you bypass the choice of Bank, default values will be assigned. If you had a machine language program called TEST, with a starting address of 7168 and end address of 8344, your save command might look like this:

↑TE "TEST", 7168,8345,B1,S14

The B must be 0 or 1 for the only two all RAM banks in the C128, while S may be any bank from 0-15, from which the SYS call could be made.

3.3.4 Programs saved using these RAM DISK commands will appear in the *QBB* directory looking just like any other program entry, and can be called and run by the usual *ID (and RETURN). If you do not want it to run when called, your program's name should begin with the < character.

3.3.5 If RAM DISK is activated, any program may be erased from the *QBB* by simply entering #ID (and RETURN).

3.3.6 If you have chosen the Save/Replace option, the top 16K bank in the *QBB* will be used to provide protected storage space for any Basic program which will fit. A corresponding 16K will be subtracted from the storage available to the Manager and RAM DISK. For this reason, Save/Replace is not availabl for 16K versions of the *QBB*. When Save /Replace is installed, the up arrow ↑ alone is all that is necessary to move a Basic program from normal RAM to the *QBB*. To return the program to normal Basic space, the left arrow is used. This recalls, but does not run the program. This option allows the programmer to save constant updates. Each new save using the ↑ alone replaces the con-

tents of the *QBB*'s top bank with the new Basic program. Programs which are too long for the 16K space will generate an overflow error message. If the ↑ is used with an ID and "NAME" the save will be to the main part of the *QBB*, with an entry in the directory. Since it requires a full 16K of memory beyond what the Manager uses, the Save/Replace is not available for 16K versions of the *QBB*.

3.3.7 To aid in remembering which options were specified when a *QBB* is initialized, if the RAM DISK is activated, the up-arrow and # will be displayed in the title line of the *QBB* Directory. If you choose SAVE/REPLACE, the back arrow will be included as well.

3.4 AUTO BOOT

The Auto-boot program is shown as the entry *AB = AUTOBT in the *QBB* directory. When the Auto Boot runs, it attempts to load and run the first program on the disk in your drive. It is particularly useful if you want a disk program to load and run automatically when the computer is first turned on. To activate the Auto-Boot feature each time the computer is turned on or reset, you can use the Manager A command to designate AUTOBT as the Auto-Start program. If you don't wish the program to load and run, you may abort that by holding down the CONTROL key as the computer is turned on or RESET. The Auto-boot is only used in the C64 mode, since the C128 has this feature built in.

3.5 RELOADING THE CARTRIDGE

When you wish to change the entire contents of the cartridge by reloading it with a previously saved "BOX", you must use the *QBB Loader* program. This should be on disk, as "QBB LOADER". This utility should be loaded with the *QBB* slide switch in the 128 position, but RUN with the switch in the 64 position. When using a C128 computer, it must be in the 64 mode, which would normally be done by powering up the computer while holding down the Commodore key. When this has been done, enter LOAD "QBB LOADER",8 and after the disk action has accomplished this, put the slide switch in the 64 position and enter RUN. This will quickly cause the screen to ask for the name of the box file you wish to load, which you enter without quotes. As soon as the disk access is complete, push the RESET (with a 128 cartridge put the slide switch in the 128 position before resetting). The screen should then show the box Directory.

3.6 LINKING PROGRAMS USING + SIGN

Several programs may be linked if a + sign is used as the first character of each name. Load each of the programs into the *QBB* separately using the Manager or the up arrow ↑ command as described in Sec. 3.3. Give each program a different ID and name, but all of the names should use the + sign as the first character. Then write a short Basic program that will be used to load and run these programs. If, for example, you want to call and run three programs that are listed in the *QBB* directory as:

```
*TE = +TELPN
*BC = +BRCLT
*TR = +TRDNG
```

your program might look like this:

```
20 *TE       : REM call program TE
30 *BC       : REM call program BC
40 *TR       : REM call program TR
50 SYS nnnn : REM start address of first program
```

You can omit the SYS statement if you use a normal name for the program which starts things running. Make it the last of the programs that you call with your short Basic program, and when it is called it will load and run without needing a SYS call. Often programs you will type in from magazines will call several programs from a Basic start up program. You can alter the start up program by replacing the Basic load commands with your own *ID call. Then load the start up program into the *QBB* and call it when you want to load and run the linked programs.

3.7 CHECK SUM

Normally, the Read Only mode should keep the *QBB* contents inviolate. However, there are some hazzards to the box contents, such as static electricity, inadvertant re-programming, or a crash due to having two cartridges turned on at the same time on an extender card. The CHECK SUM utility is a small program that adds up all the byte values in the *QBB* and prints a total on the screen. Its usefulness is predicated on the fact that any change in the box contents would have a high likelihood of changing the total. It is called by entering *H (RETURN). The version supplied with the *QBB* only works on C64 boxes.

3.7.1 When you have the *QBB* filled pretty much as you would like to keep it, and do not expect any of its contents to change, it only requires one added block to load in CHKSUM. Once it is ready, call it and note the result on a small tag pasted to the box. Then, whenever you wish to be reassured that the box contents haven't changed, just enter *H then press RETURN and the checksum number should again be displayed on the screen.

3.7.2 You may find that when you re-load the *QBB* with a previously saved box the checksum is not what it had been the last time you had that particular box loaded. The probable reason for this is that the checksum totals the whole cartridge, and the box load usually does not fill every byte, so a previous box usage may have changed some of the residual bytes in the upper part of the cartridge. Thus, the checksum is not a verification of a previous box, but just assurance your present box hasn't changed.

4.0 PROGRAMMING THE QBB

There are many applications for a *QUICK BROWN BOX* in addition to saving and running pre-written programs using a *QBB* Manager utility. Some of these might include: storage of data; writing your own Auto-Start application; using a portion of the box as non-volatile, safe RAM for entering or writing programs without having to do frequent backups; and opening up space for programs too large to fit in presently available RAM. These applications are too complicated for simple Managers to handle. Our Manager utililties save the user from having to deal with a lot of programming details in the *QBB*. However, to develop other applications you will need to know what goes on under the hood.

4.0.1 The *QBB* consists of 16K banks of battery backed CMOS memory. These banks are numbered 0 and 1 (for 32K boxes) and 0 thru 3 (for 64K boxes). Only one bank may be on at a time, and they all use the same address range, $8000 to $BFFF (Decimal 32768 to 49151). The *QBB* is controlled by a Write Only control register whose address is $DE00 (Decimal 56832). The register controls several functions, including the selection of a particular 16K bank and the selection of the Read Only or Read/Write mode. These functions are listed in the table below, and are described more fully in Sec. 8.2. Each function is given a numerical value. The numerical value assigned to placing the *QBB* in the Read/Write mode, for example, is 16, and the value for selecting bank 3 is 3, so to load a program or write data into bank 3 would require 16 + 3, or a value of 19 to be poked into address 56832. For the convenience of the computer and *QBB* hardware, the numerical values of the operating functions are related to certain 'bits' of the Control Register. It is not essential for programming the *QBB*, but may be useful at times to know that these bits are numbered 0,1,2,4,6, and 7. When a control function is to be turned on, its related bit (or bits) is set to a logic high. The control functions are summarized in the following table, listing the bits and the numerical values of each bit (or combination) in both decimal and hexadecimal units.

FUNCTION	BITS HIGH	DECIMAL	HEX
Select Bank 0	None	0	0
Select Bank 1	0	1	1
Select Bank 2	1	2	2
Select Bank 3	0 & 1	3	3
Pull /GAME line LOW	2	4	4
Select Read/Write	4	16	$10
Pull /EXROM LOW	6	64	$40
Disconnect *QBB*	7	128	$80

4.0.2 A hardware RESET or power up zeros all bits. That leaves the *QBB* connected, with bank 0 selected, in Read Only mode, and neither /GAME or /EXROM is pulled low. The one other control on the *QBB* is the slide switch, which in the 64 position pulls /EXROM low regardless of the setting of bit 6 of the Control register. With /EXROM low, but /GAME high, the *QBB* takes over the 8K segment of memory space between 32768 and 40960 ($8000 to $A000). For a C64 in this mode, the upper half of the *QBB* address space cannot be read because the Commodore Basic ROM is in control from $A000 to $BFFF. If bit 2 is activated (set high) thus causing /GAME to be pulled low, you will be able to read the upper 8K of the *QBB* bank, but you will not be able to use Basic. With the *QBB* plugged into a C128, and slide switch in 128 position, all 16K of the *QBB* bank is accessible.

4.0.3 There is no way to read which bits of the control register are set or cleared, so it is up to the programmer to keep track of all six of the control bits, so that each write operation to $DE00 will invoke the desired operating functions.

4.0.4 In order to do any programming in the box, you must first put the box in the Read /Write mode. This can be done from the Basic Immediate mode by a POKE 56832,16. A machine language routine to do the same thing would be:

```
LDA #$10
STA $DE00
```

The more useful control register ($DE00, 56832) modes are summarized in the following table, which applies to bank 0:

64 MODE	FUNCTION	HEX	DEC	(Poke)
8K	Read Only $8000-9FFF	0	0	RESET
16K	Read Only $8000-BFFF	$04	4	NO BASIC
8K	R, 16K W	$10	16	
16K	Rd/Write $8000-BFFF	$14	20	NO BASIC
Disconnect Box		$80	128	
128 MODE	**FUNCTION**			
16K	Read Only $8000-BFFF	00	0	RESET
16K	Read/Write	$10	16	
Disconnect Box		$80	128	

The corresponding HEX or Decimal values for banks 1-3 are set by adding the bank number to any value in the above table. For example, to Read or Write a full 16K of bank 0 in the C64, the table shows that $14 must be loaded into $DE00. The same access to bank 1 would require writing $15, or for bank 3, $17 into $DE00. The only values that are insensitive to the bank values are the settings to disconnect the box.

4.0.5 Since the *QBB* runs as part of the C64 or C128, a programmer could, in theory, program the cartridge in any language that is presently available for use in these computers. In practice, the fixed address space of the cartridge, $8000 to $BFFF, may not coincide with the work space allotted to the programmer by currently available implementations of Pascal, Forth, C, Logo, or Comal. Thus, we must restrict our consideration to BASIC and Machine Language. BASIC is native to these Commodore computers, a machine language monitor is part of the C128, and we have included a machine language monitor, Supermon, in the cartridge for the C64. Some machine language assemblers, such as PAL or MAE adapt easily to the *QBB* environment so that the assembler itself and a large work space can reside simultaneously in the cartridge.

4.0.6 First, you must decide how much of the box you are going to set aside for your application, and how much you will reserve for loading programs with a *QBB* Manager. This decision is made when you are first initializing the *QBB*, as discussed in paragraph 3.1.7. Besides the partitioning described in par. 3.1.9, you could pretend that a 32K box is really only 16K, or a 64K box is really 32 or 16K when you are asked to choose its size when initializing. The highest 16K bank or banks can then be used any way you like. For a 64K *QBB* that has been told it is 32K you would call bank 2 either in the Basic immediate mode with a POKE, or from a machine language program you had already written to run outside of the *QBB* address space. You do have to exercise some caution, since code in one bank which calls another bank disappears when the bank changes. This may seem somewhat obvious, but it is quite pertinent when you are moving data or have programs that cross bank boundries.

4.0.7 Some applications may prefer to have the *QBB* disappear for a while, so that the RAM or ROM underneath can be used. For the C128, this is done by 128 bank switching which allow either RAM or Basic ROM under the cartridge to be read. Just remember that in order to do anything to the *QBB*, the I/O bank at $DE00 must be available. The C64

banking is set by /EXROM and /GAME (bits 6 & 2) of the *QBB* control register and bit 0 of RAM location 1. The slide switch overrides bit 6 in 64 position. More about this in section 4.3.

4.1 PROGRAMMING IN C-64 MODE

The most important consideration when programming directly in the *QBB* is the amount of space you expect the program or data to occupy. As the table in 4.0.4 indicates, at any one time you have access to either 8K or 16 Kilobytes of the box. The section from $8000 to $9FFF is available for any thing you wish to put into it once the Read /Write bit is set. The upper half of a *QBB* bank is addressed $A000 to $BFFF, which is also the address of the Basic ROM. In order for you to write or run a Basic program, the ROM must be available so your Basic program may only use the lower half of the *QBB*. You would POKE 56832, 19 to make the lower half of the bank 3 available (see next par.). A POKE of 23 would give you Read/Write access to the whole 16K if you choose to use machine language, and assuming you will not be using any subroutines from the Basic ROM.

4.2 It is possible to write or load Basic programs up to 8K in size directly into *QBB* memory and run them in the *QBB*. The Basic pointers (low memory $2B to $38) which normally assign the beginning of Basic memory to $0800 (2048) can be altered to start Basic memory at $8000 (32768) which coincides with *QBB* memory. The pointers can be changed by poking new values. This changing of the pointers can get a little sticky, but there have been a number of articles published on creating multiple Basic programs that handle the pointers. Remember, the box must be kept in the Read/Write mode when a Basic program in it is running. It is important to note that some Basic programs which seem to fit nicely into the an 8K space may not run well because they require an unusual amount of space for strings and variables.

4.3 In some applications it may be desired to have access to the C64 RAM under the $8000 area of the *QBB*. Be aware that writing to the *QBB* always causes the same code to be written to any RAM underneath.

4.3.1 It is simple to write to the RAM under the *QBB*, just keep the *QBB* in Read Only, and load or write the RAM as though the *QBB* were not there. If what you are loading depends on the location of the top of Basic, you will have to put the slide switch in the 128 position and do a software RESET by SYS 64738.

4.3.2 Reading back what you have written is a little more involved, because normally, with /EXROM pulled LOW (as it is with the slide switch in 64 position), you read a cartridge from $8000 to $9FFF. You may clear bit 0 of address location 1 to gain access to all of the RAM under the cartridge, from $8000 to $BFFF. This, of course, disables Basic. To read just the RAM from $8000 to $9FFF, leave bit 0 of location 1 set, and put /EXROM HIGH. This may be established under program control if first you set bit 6 to hold /EXROM LOW, then have the program stop and ask the user to put the slide switch in 128 position. Once this has been done, the program now can have full control of whether RAM or *QBB* is selected, just by the programming of bit 6. This bit control of the $8000 to $9FFF bank may be kept through a SYS 64738 RESET as described in section 6.3.

4.4 Another somewhat complex use of the *QUICK BROWN BOX* is the AUTO START mode. The *QBB* can be programmed to take control of the computer upon power up and execute a machine language program prior to (or instead of) entering the normal Basic monitor. The AUTO START section (6.0 and 6.1) describes how this may be done.

4.5 WARNING. The *QBB* uses a portion of RAM that is normally allotted to Basic. When it is plugged in there are only 30719 bytes in the regular Basic space instead of the usual 38911 bytes. This may not be enough for some programs to run while the *QBB* is in place. One solution is to switch the *QBB* to the 128 mode, and then push the RESET. A program can do the same thing by writing a one to bit 7 of the *QBB* control register, but be careful to restore the Basic error vector $E38B at $0300 and $0301 because many *QBB* utilities alter it to point to the box, which would be bad news if the box were disabled. In Basic Immediate mode this would be POKE 768,139 : POKE 769,227.

5.0 PROGRAMMING THE QBB (C-128 MODE)

A problem when using the *QBB* in the 128 mode is that the Basic 7.0 ROM uses the same address space. There is, therefore, no simple way to write or run a Basic program in the box. You can, of course, write a Basic program in normal Basic space and store it in the box, described in Sections 3.2 and 3.3. We are left then with machine language programming, for which the box is well suited.

5.1 As you must know, if you are going to tackle M.L. programming in the C128, this computer is designed to allow partitioning into 16 so-called 'banks'. The bank of greatest interest when using the *QBB* is bank $D (13), because it is the only bank that allows the cartridge to co-exist with the Kernal.

5.1.1 As mentioned previously, before you can begin to build a program in the *QBB*, you need to have the box in the Read/Write mode. You can write the appropriate bits to the *QBB* control register ($DE00) in any bank except banks 0-3 and 14. Be aware that certain *QBB* manipulations use the first 12 bytes in the $C00 page. It is often more convenient to have written a few short routines that can call the desired *QBB* mode. The cassette buffer at $0B00 is a good place to store such routines. After calling the 128 MONITOR you could assemble the following:

```
A FOB00 LDA #00   ; READ ONLY
  FOB02 STA DE00
  FOB05 RTS
  FOB06 LDA #10   ; READ/WRITE
  FOB08 STA DE00
  FOB0B RTS
  FOB0C LDA #80   ; DISCONNECT
  FOB0E STA DE00
  FOB11 RTS
```

Once these are assembled, if you wish to load or write into the box, enter (from the MONITOR) J FOB06 and that will put it in the R/W mode. J FOB00 would put it in Read Only, though a RESET would do that also.

5.1.2 When the box has been placed in R/W you can begin your assembly to any address in the box by using that address preceded by a D. If you want to be able to use Kernal routines in your program, you would

early in the program have it establish the configuration for Bank 13 by writing:

LDA #0A, STA FF00

5.2 There are several excellent books that you may wish to consult for more information about programming the C128. Three of the best are:
COMMODORE 128 INTERNALS published by Abacus Software.
COMMODORE 128 PROGRAMMING SECRETS by Wiese, publ. Osborne McGraw-Hill
COMPUTE!s 128 PROGRAMMER'S GUIDE published by COMPUTE! Publications.

5.3 When you read about the 128 Kernal routines such as JMPFAR, JSRFAR and FETCH that were designed to simplify cross-bank code manipulations, you will find that they can all be useful to you in the *QBB*. However, be aware that all these routines were designed to be used from Bank 15. You may need to make minor changes to run a routine from Bank 13.

5.3.1 JSRFAR, for example, is set to return to Bank 15. When you use it from bank 13, you will do the normal set-up but in addition you will need to add the following code:

```
LDA #0A     ; Configuration for Bank 13
 STA  2DE       ;  Page  2  address  holds  return
configuration
JSR FF6E    ; JSR JSRFAR
LDA #00     ; Configuration for Bank 15
STA 2DE     ; Restore previous Config.
```

On occasion, it is helpful to disable the interrupts before crossing banks, then restore interrupts when you return.

5.4 The C128 has a FAST mode that allows the microprocessor to run at a 2 MHz rate, twice as fast as the normal speed of execution. Unfortunately, for their own obscure reasons, Commodore did not allow any 'Bank' using a cartridge to operate in the FAST mode. Thus, a program may be stored in the *QBB* and when it is to run it can be moved to regular RAM and use the FAST mode. If the program is to run within the *QBB*, it may only run at the normal speed.

5.5 There are a greater variety of machine language assemblers available for the C64, and if one of these is available, it is perfectly feasible to write the actual code using the box in the 64 mode even though it will be run in the 128 mode.

6.0 AUTO-START, 64 MODE

The first storage device to get a crack at having its program contents run when the computer is turned on is a cartridge. Upon power-up, or a RESET, the computer checks for the presence of a cartridge before it attempts to do any other initialization. It does this by looking to see if locations $8004 through $8008 hold the consecutive bytes C3 C2 CD 38 30. If you put these bytes into the *QUICK BROWN BOX* at that address the computer will transfer control by a jump indirect to whatever address is specified at locations $8000, $8001 in low byte-high byte order. Usually that address will be $8009, so 09 will be in $8000, and 80 in $8001, and your program would be set to begin at $8009. The processor status conditions in effect at that point are Decimal mode cleared and Interrupt Disable set.

6.0.1 When running an Auto-Start program, if it crashes and hangs up

the computer, just put the slide switch in the 128 position and press RESET. The computer will have a normal Basic start up, from which you may load a machine language monitor and then begin to fix the problem.

6.1 CUSTOM AUTO START

When you program the Auto-Start mode, you are really letting the computer know that you are the boss! You may wish to begin with just a modest assertion of your authority, such as choosing your own colors for characters and background, after which you could give control back. To do this in the easiest manner, your cartridge routine would have to duplicate some of the computer's initialization routine
Relevant *QBB* code begins:

```
$802E JSR $FDA3    ;Initialize CIA I/O
$8031 JSR $FD50    ;RAM test, set pointers
$8034 JSR $FD15    ;Set vectors for KERNAL
$8037 JSR $FF5B    ;Init. Video Controller
$803A CLI          ;KERNAL init. finished
$803B JSR $E453    ;Set Basic vectors
$803E JSR $E3BF    ;Initialize zero page
```

At this point you can insert a jump to subroutine call to your own program. If you don't need to use Basic, the call to your program can be a JMP. If Basic is to be called, as it is in the *QBB*, the following additional steps initialize Basic without undoing whatever set-up had already been done in your program.

```
$8041 JSR $A644    ;does CLR and NEW
$8044 JMP $E394    ;gives control to Basic.
```

6.2 RESTORE KEY

The RESTORE key initiates a Non-Maskable Interrupt which will vector directly to the cartridge, and if the bytes at $8004-$8008 call the Auto-Start mode, control will pass to the location held at $8002,8003.

6.3 RESET

Under certain circumstances it may be desireable to hold the *QBB* in a mode different from its normal start up mode while resetting the computer. This cannot be done using the RESET button, because that would reset the *QBB* control register. A SYS 64738 cold start call would work, since it sends the computer through all of its initialization, but doesn't affect the *QBB*.

7.0 AUTO-START 128

The AUTO-START mode for the C128 is similar to that of the C64. On power-up or RESET, the computer looks for a sequence of bytes that indicate the presence of a C128 cartridge. These bytes are the ASCII code for CBM and the computer looks for them starting at $8007. Unlike the C64 routine, the most significant bits of the letter bytes are not set. A more significant difference, however, is that you have a choice that is not available in the C64 mode. The byte at $8006 is a special identifier. If it is 01, the computer immediately yields control to the cartridge and jumps through the cartridge cold start vector, which in another departure from the 64, is a JMP direct from $8000 at which the code $4C L.B. H.B. will have been written. If you want to exercise this option, you will have to take over the rest of the computer initialization, as you do for the 64. If you would like to avoid taking on the responsibility for completing the initialization, you place any

number greater than 1 and smaller than $FF in location $8006. In this case, the computer will note the presence of the cartridge, complete its own initialization, and then return to the cartridge to execute your code, taking its JMP from the warm start vector located at $8003. This is the system we use for all the 128 cartridges:

```
$8000 4C 0A 80 JMP $800A  ; Could be omitted
$8003 4C 0A 80 JMP $800A  ; Warm Start Vector
$8006 11                  ; Identifier byte
$8007 43 42 4D            ; "CBM"
$800A   Our code begins here.
```

It is important, if you wish to use Kernal routines in your own machine language program, to establish the configuration for Bank 13 immediately. The computer, when it identified the cartridge, assumed it was in Bank 8. Therefore our first instruction at $800A is always:

```
$800A LDA #0A
$800C STA FF00
```

From this point on you are free to experiment on your own. If your efforts are not completely successful, a crash in the 128 mode is not a disaster. Flip the slide switch to the 64 position and press RESET. You can then load a monitor or assembler and make changes to your code, put the switch back to 128 position, and try again.

8.0 THEORY OF OPERATION

The important operating modes of the *QBB* are selected by the slide switch and a control register whose address is $DE00 (56832). This is a Write Only register, using bits 0,1,2,4,6 and 7. Don't attempt to PEEK the register, as it could leave the register loaded with indeterminate commands. The slide switch pulls /EXROM LOW when in the 64 position, and leaves it to the discretion of bit 6 when in the 128 position. This allows more flexible control when in the C64 mode (see section 4.3).

8.1 When the computer is turned on, or the RESET button is pushed, the control register is reset to a pre-determined mode. If the slide switch is in the C64 position, the *QBB* comes on in the Read Only mode, with just the lower half ($8000 to $9FFF) accessible so that the normal computer Basic will run. In the C128 position, the box still comes on in the Read Only mode, but all 16K (from $8000 to $BFFF) are accessible. In the C128 computer Basic and external cartridges have to compete for the same address space so special procedures are necessary to gain access to the *QBB*. These are described in Section 5.

8.2 The Control Register is a write only register, using six of the eight available data bits. They must all be written in each write operation, so it is necessary to keep track of what bits are already set when you are ready to make a change, and make sure the new word affects only the bit (or bits) you wish to change. A summary of the bit values required for each operating mode is contained in paragraph 4.0.1. The effects of each bit are as follows:

BIT 7 if a HIGH (1) disables the *QBB* memory (but not the control register) so that just about any other program or cartridge can run, just as if the *QBB* were not plugged in. An important exception to this is any cartridge, such as some Fast Load and special copying cartridges that also use $DE00 (56832) as a control register. A zero at this bit turns the *QBB* back on. Thus, if you are in Basic, a POKE 56832,128

would turn the cartridge OFF and POKE 56832,0 turns it back on.

BIT 6 when HIGH causes the /EXROM line to be pulled LOW. The /EXROM line is the one that must be pulled LOW to cause the computer to know that a C64 cartridge is plugged in, and the slide switch already holds /EXROM LOW when in the C64 position. Thus this bit will only have an effect if the switch is in the C128 position, and if set to a 1 will force the C128 computer into the C64 mode if SYS 65341, the warm RESET is entered. The reverse, writing a zero to this bit will not bring the computer back in to the 128 mode. The only way to accomplish that without destroying whatever is stored in RAM is to push the RESET button.

BIT 4 controls whether the *QBB* acts as RAM (Read /Write mode) or ROM (Read Only mode). A HIGH (1) selects R/W and 0 selects RO. Whenever you have finished programming with the *QBB*, it is important to make sure the box is left in the Read Only mode, that is, with this bit cleared.

BIT 2 controls the /GAME line, the other C64 cartridge control line. When it is brought HIGH it causes /GAME to go LOW. It would not normally be used in the C128 mode. If the slide switch is in the C64 position, or if bit 6 is HIGH, causing /EXROM to be LOW, and you then cause /GAME to also go LOW, Basic will be disabled, and the upper 8K of the *QBB* ($A000-$BFFF) can be read. If you just wish to write to that part of the *QBB*, it doesn't matter whether /GAME is asserted or not, but the cartridge does have to be in the Read/Write mode.

BITS 1 and 0 control the selection of the 16K banks in the *QBB*. Bit 1 is only used in the 64K box, so if an attempt is made to call bank 3 (both bits HIGH) the 32K *QBB* will turn on bank 1.

8.2.1 In the C128 mode the /EXROM and /GAME lines are not used, but they can be set high or low. The *QBB* was designed so that it would not interfere with possible future uses of /EXROM and /GAME. Therefore its outputs to these lines are open collectors. Thus when bits 2 and 6 are low, those lines see an open circuit.

8.3 The /DE00 input to the cartridge writes into the control register whenever an address from $DE00 to $DEFF is programmed. Since we have not provided a read lockout, a READ to $DE00 will attempt to write random garbage to the control register. Therefore, any attempt to read the register value, which is impossible anyway, will instead write indeterminate data into it. Treat it as a WRITE ONLY register.

8.4 When the *QBB* is plugged into the computer, and the computer is turned on, there is no drain on the internal 3 volt lithium battery. When the computer is off, or the cartridge un-plugged, the drain is about 1 microampere. The battery is rated at 160 milliampere-hours, so assuming the worst case, the battery should last its expected shelf life of 10 years.

9.0 OPERATION with EXPANSION RAM & Other Cartridges

CAUTION IF YOU USE THE QBB ON AN EXTENDER CARD:
There are a number of extender cards available that allow several

cartridges to be plugged into your C64 or C128 at the same time. Most of these have individual slot switches that allow you to turn on or off each cartridge individually. If you have the *QBB* on at the same time as another cartridge, you run a risk of corrupting the contents of the *QBB*, though we have never had an instance where this mishap caused any permanent damage to the box. Thus, be sure to back up your box on disk or tape before using it with other boxes on an extender.

9.1 If a *QBB* and one or more other cartridges are connected to the expansion port at the same time, and have power applied, they each may try to place data on the eight lines of the Data Bus. Since some data lines will be trying to go HIGH while others will pull LOW, the odds are that at any one time, half of the lines are in concert, and half will be in contention. Those in contention will draw excessive current, and therefore cause the corruption of the data in the *QBB*.

9.1.1 If bit 7 is HIGH in the *QBB* control register, it keeps the cartridge from placing data on the Data Bus, so even if powered up at the same time as another cartridge, the *QBB* data should be safe and protected from any contention problems.

9.2 Another problem may arise from other cartridges which also use $DE00 as a register address. Even if the *QBB* control register has bit 7 set, if the computer were attempting to write the control register of another cartridge, that data would also be written into the *QBB* control register, perhaps putting it in the Read/Write mode, or clearing bit 7 and allowing bus contention.

9.3 The *QBB* was designed to be compatible with the Commodore 1700, 1750, and 1764 RAM Expansion Units. These units use $DF00 as their control address, so control commands will not interact.

9.4 It is not a good practice to leave the *QBB* plugged into an expander card for extended periods of time when the computer is on, but with the slot switch for the *QBB* turned off. This causes a drain on the internal lithium battery about 25 times higher than normal. When the computer is turned OFF, there is no such problem.

9.4.1 When using the *QBB* on an expander with the C64, if the slide switch is placed in the 128 position, the *QBB* could have its slot left powered up and be invisible to most other cartridges. This does have problems if another cartridge uses I/O addressing in the $DE00 page. It also places 40 or 50 milliamps additional load on what is usually, on the C64, a marginal power supply at best.

10.0 SUPERMON+ COMMANDS

The original version of this monitor by Jim Butterfield is relocatable, but we chose to include a version that loads at $C527. Once loaded by entering *S, it may be called from Basic by SYS 50471.

NUMBER BASE CONVERSION $ + & %

R REGISTER DISPLAY

M MEMORY DISPLAY (With ASCII)

```
X EXIT TO BASIC MONITOR

D DISASSEMBLY    D hhhh

A ASSEMBLY   A aaaa LDA hhhh

F FILL MEMORY  F ssss ffff bb

G GO (Begin to run program) G ssss

J JUMP TO SUBROUTINE        J ssss

H HUNT   H ssss ffff 'word
         H ssss ffff bb bb bb bb (32 max.)

L LOAD   L "FILENAME",08  or ,01 (Disk or Tape)

S SAVE   S "FILENAME",08,ssss,ffff+1

T TRANSFER   T ssss ffff nnnn

C COMPARE (memory) C ssss ffff nnnn

DISK COMMANDS  @ Gets disk status message
               @9                  Gets unit 9 status
               @,$0 (number zero)  Displays Dir. Dr. 0
               @,S0:filename       Deletes that file from disk
               @,N0:filename,ID    Formats disk
               @,I                 Initializes disk
               @,V                 Validates disk
    SYS 50471 Calls QBB version after it is in place.
```

11.0 PROBLEMS AND TROUBLESHOOTING

We may divide any possible problems into three catagories: 1. *QBB* faulty when you received it; 2. *QBB* was good, but something went wrong immediately after you performed some action on the box or ran a new program; and 3. *QBB* was good, but went bad spontaneously, with no help from you. For the first and third cases, perform the tests outlined in Section 11.1.

11.0.1 If you are not sure whether the problem is something you did, or a true fault with the box, the first thing to do is to try to restore the original box contents. If you have saved a progam file of about 5 blocks called *QBB* LOADER and a sequential file over a 100 blocks long named whatever you named it, just follow the instructions in Section 3.5. When the loading is done, when you press the Reset, you should see the original box directory. The checksum will probably not be the same as it originally was.

11.0.2 If you are attempting to set up a C128 box, you MUST save M8 to disk as described in Par. 2.2.3, then put the slide switch in 128 position, Reset the computer, load the 128MANAGER, and INITIALIZE the cartridge.

11.1 HARDWARE CHECKUPS

We must first determine whether the *QBB* circuitry seems to be

intact. If you have saved it to disk, run the *QBB* CHECKER. This is run with the slide switch in the 64 position, even if testing a box loaded with C128 programs. It verifies that the *QBB* memory and control register are functional. If you don't have the checker, the following steps will give a limited verification.

A. With the computer OFF, plug in the *QBB*, (or just turn it OFF if already plugged in).
B. Place the Slide Switch in the 128 position.
C. Turn on the computer in 64 Mode. If your computer is a C128, hold down the C= key when powering up to force the 64 mode.
D. The normal Commodore 64 screen display with 38911 bytes free should be visible, along with READY and a flashing cursor. If not, the cartridge is bad, and must be sent back for repair or replacement. Further testing at this point would be useless.
E. Enter the following two lines, each followed by a RETURN:

```
10 FOR A = 0 TO 3 : POKE 56832,A+16
20 POKE 32772,A : NEXT : POKE 56832,0
```

F. Check that what you have typed agrees exactly with this manual, then enter RUN (and RETURN). This should write the bank number into the fifth byte of each bank.
G. Turn the computer OFF for at least 30 seconds.
H. Place the Slide Switch in the 64 position.
I. Turn the computer ON.
J. The normal Commodore 64 message, with 30719 bytes free should be visible. If it is not, or it reads 38911 bytes, send it back to BROWN BOXES.
K. Enter the following two lines, each again followed by RETURN:

```
10 FOR A = 0 TO 3 : POKE 56832,A
20 ?PEEK(32772) : NEXT : POKE 56832,0
```

L. Again check for accuracy then enter RUN (and RETURN). A column of four numbers should be printed on the screen: 0,1,2,3 for a 64K *QBB*; 0,1,X,Y for a 32K box; or 3,3,3,3 for a 16K box. X and Y could be any random numbers, but if the specified numbers are not all correct, the box is bad.
M. Enter the following line (followed by a RETURN):

```
POKE 32772,255 : ?PEEK(32772)
```

N. The screen should show a 0, indicating that the box is in Read Only, as it should be. If it shows any other number, it's bad.

11.1.1 If the *QBB* passes the tests of the previous paragraph, there is an excellent chance it can be loaded with its previous contents. If you have been able to save the original box contents and the QBBLOADER, reload it using the instructions in Section 3.5. If this is unsuccessful, or you never had the chance to save the original contents, notify BROWN BOXES that the contents of your box seem to have been corrupted, and you will be sent a disk that contains the original box contents.

11.1.2 If you wish to return the box to its exact condition before you conducted the above tests, press RESET then enter the following:

```
10 FOR A = 0 TO 3 : POKE 56832,16+A
20 POKE 32772,195 : NEXT : POKE 56832,0
RUN
```

11.2 OPERATIONAL PROBLEMS

11.2.1 PROGRAM WON'T LOAD INTO QBB.

A. When program name is entered, disk drive is active only an instant, and nothing is in *QBB* directory. Probable Reason: It's a copy protected program.

B. Same, except a 1 or 2 block entry is in *QBB* directory. Probable Reason: the main program has a different name and is loaded by the short program you put into the *QBB*. This short program would use disk commands to call the main program, and is often written in Basic. If so, it is usually a simple matter to designate the main program as a linked file, and substitute *ID for the disk call. Refer to Sec. 3.6.

C. FILE TYPE MISMATCH error message is received. Probable Reason: You are attempting to load a Sequential file. The Managers only handle Program files. Data files in general require different loading utilities, and we havn't written them yet.

11.2.2 PROGRAM WON'T RUN WHEN CALLED

We assume that the program loaded into the *QBB* and showed the same number of blocks in the *QBB* directory as on the disk.

A. Some C64 programs get put on to disks in funny ways. Save to *QBB* again using this method:

1. Load program from disk into computer.
2. Immediately SAVE it back to disk, under a different name.
3. Load C64 Manager and delete the earlier version from the box, then load the *QBB* the with newly saved version.

B. Some C128 programs object to the GRAPHICS mode. If that mode might have been set prior to calling the new program from the *QBB*, RESET the computer and try again.

C. Many C64 programs use the same address space that the *QBB* uses. If RUN while *QBB* is ON, the program will crash. Reload program into *QBB* with its six character name beginning with the <. Then when you call the program it loads, but doesn't RUN. You now have time to turn OFF the *QBB* by putting the Slide Switch in the 128 position before running the program.

D. Some data or text files are stored on disk as Program files. These are not meant to run.

E. Some C128 programs have been given the wrong starting address when sent to disk. There isn't much that can be done unless you are adept with a Disk Doctor and can alter the address directly on the disk, then reload into the *QBB*.

F. Some C128 programs make use of memory area near the low part of $C00 page, which is also used by the *QBB*. If such a program ends with a return to Basic, or requires further access to the *QBB*, a crash would occur, necessitating a RESET. There is not much that can be done to resolve this conflict except to revise the program or the *QBB* Manager to use a different area.

INDEX **PARAGRAPH**

ASSEMBLER..... 4.0.5 5.5
AUTO-BOOT..... 3.4
AUTO-START..... 3.1 3.1.6 6.0 7.0

BANKS C-128..... 3.2.1 3.3.3 5.1
  Change 3.2.2
  QBB 4.0.1 4.0.4
BASIC
  Box storage..... 3.3 4.2, 4.5
  Disable 4.0.2 4.1
  Error Vector 4.5
  Pointers ($2B-$38) 4.2
  Save/Replace 3.3 3.3.6
BATTERY..... 8.4 9.4
BOOKS on C128 ..... 5.2

CHECK SUM..... 2.1.2 3.7
COMMERCIAL SOFTWARE 2.0.2 3.0
CONTROL KEY..... 3.2.5 3.4
CONTROL REGISTER ..... 4.0 - 4.0.4
  Bit Tables..... 4.0.1 4.0.4
  Operation 8.2
  Progrmming 5.1.1

DELETE 2.2.4 3.1.5 3.1.6 3.3.5
DISCONNECT BOX 4.0.7 4.3, 8.2

ERROR VECTOR 4.5

FAST MODE (C128) 5.4

INITIALIZE 3.1.3 3.2
INTERRUPT, Non-maskable 6.2

KERNAL 5.1.2 5.3 7.0

LINKED PROGRAMS 3.2.8 3.6
LOADER UTILITY 2.2.2 3.1.7 3.5

MACHINE LANGUAGE 4.1.1 4.2 5.0
MALFUNCTIONS 9.2 11.0-11.2
MANAGER C64..... 2.2.1 3.1
C128..... 2.2.4 3.2

PARTITIONING..... 3.1.10 4.0.6
PROGRAM DESCRIPTION 3.1.9 3.2.2
PROGRAM SIZE, NUMBER 2.0.2 3.1.5
POINTERS 4.3

RAM DISK 3.1.3 3.3
RELOADING BOX CONTENTS 3.1.1 3.5
RESET 3.5 4.0.2 6.0.1 6.3 7.0 8.2
RESET BUTTON 2.1.1 2.2.2 2.2.4
RESET, SOFT..... 4.3.2 6.3
RESTORE Key 3.1.6 6.2
RUN (and RUN disable)..... 3.1 3.1.8

SAVE TO QBB..... 3.3.2
SAVE TO DISK..... 2.2.3 2.2.4 2.2.7 3.2.3
SEQUENTIAL FILE 3.1.7
STATIC ELECTRICITY 2.1
SUPERMON (Location $C527-$CFFF) 10.0
SWITCH, Slide 2.1 2.2.3 3.5 8.0 8.1